Remarks

Claims 1 - 81 are presented for reconsideration. Claims 1, 6, 7, 15, 20, 21, 29, 35, 36, 47, 54, 55, 59, 60, 65, 74, and 75 have been amended.

References to the Specification are made in accordance with the paragraph numbers of the publication hereof in Publication No. 2003/0014734 A1.

Objections to Claims

Claim 60 was objected to, as being a duplicate of claim 52. Claim 60 has been amended to correct its dependency.

Claim 59, although not objected to, has been amended in order to correct its dependency and to more particularly define the invention.

Rejections Under 35 U.S.C. § 112

Claims 6, 7, 20, 21, 35, 36, 54, 55, 74, and 75 have been rejected under 35 U.S.C. § 112, second paragraph. As regards the recited coverage variables, the Examiner has interpreted the language "all possible values" as being indefinite and unsupported by the Specification.

In the Specification, for example in paragraph [0141] of the Official Publication hereof, the term "possible values" has the meaning of "allowable values". The Specification offers an example of a variable Foo having possible values of {true, false}. Clearly, no other values could be assumed by the variable Foo.

Regarding claim 6, for example, a cross-product of "possible values" is recited. In paragraph [0144] of the Specification, another example is given, in which "possible values" of variables Foo and Bar are restricted to {1, 2}, and {a, b} respectively.

Applicant has amended claims 6, 7, 20, 21, 35, 36, 54, 55, 74, and 75 to limit the coverage variables to "allowable values" and to more particularly define the invention. The term allowable values, for example, in claim 6, has an antecedent basis in claim 3. Claim 3 is original disclosure, and was not objected to. Applicant believes that the amendment of claims 6, 7, 20, 21, 35, 36, 54, 55, 74, and 75 is adequately supported in the Specification in its present form. However, if the Examiner so requires, Applicant is willing to import the exact language of claim 3 into the Specification.

The Examiner appears to assert that limitations of claims 6, 7, 20, 21, 35, 36, 54, 55, 74, and 75 are found in the cited TETware documents. Applicant respectfully disagrees, for reasons that are explained in the discussion of the rejections under 35 U.S.C. § 102. Applicant thanks the Examiner for his suggestions in resolving the issues brought out in this rejection.

Rejections Under 35 U.S.C. § 102

Claims 1-81 were rejected under 35 U.S.C. § 102 as being anticipated by The Open Group's product TETware, as documented in three publications. These include the TETware User's Guide Revision 1.2 (User), the TETware Programmer's Guide Revision 1.2 (Prog), and the TETware Release Notes Release 3.3 (Notes).

Claim 1.

Independent claim 1 has been amended to claim the generation (creation) of test programs used to verify a software application responsively to the software application's own specifications.

The TETware documents disclose a set of tools for the development and execution of system and unit tests (Prog, Sec. 1.2). Test execution facilities are described extensively. However, as to test case development, which Applicant considers to mean test case generation as claimed, TETware merely provides interface libraries for building test cases into executables, the test cases having been written previously (by others) in various programming languages, e.g. C++, Perl. (User, Sec. 2.4).

The TETware product includes a case controller (tcc), which in an execution mode, is capable of executing a list of test programs which are specified in a test scenario (Prog, Sec. 2.2). The test scenario may include directives, which influence the processing or execution of the test programs, but, as explained below, have nothing to do with their creation. Applicant uses the term "generation" to mean the creation of the test cases or programs, not the routine assembly of existing test programs into executables by an execution engine. Applicant believes that the amendment of claim 1 now distinguishes the generation of executables from pre-written test programs, as disclosed by the TETware documents.

The Examiner has asserted that the TETware documents disclose generation of a test program as claimed. Applicant again disagrees, for the reasons stated above. Applicant urges that the Examiner has incorrectly interpreted the assembly of existing test programs into executables as "test generation". Applicant intends this term to mean the creation of test programs for a software application from basic specifications. The difference can be appreciated, for example, from Applicant's Fig. 1, which illustrates that the claimed foci and behavioral model provide the input to a test generator that produces abstract test suites for execution by an execution engine. These test suites are created from specifications of the software application being verified, as now claimed in amended claim 1 (Specification, paragraphs [0021], [0116], Fig. 1). The test suites are suitable for execution by an execution engine to verify the software application. Thus amended claim 1 clearly differentiates the creation or generation of test programs from the execution of test programs.

It is assumed throughout the TETware documents that the test cases it processes have been generated in advance. Indeed, this is made explicit (Prog, Sec. 2.4.3), wherein it is stated:

> "When you write a test case which uses a TETware API, you only need to supply the test purpose code that actually performs the required test operation. When a test case is executed, the TETware TCM calls each test purpose function that you write and ensures that each test purpose registers exactly one test result." [emphasis supplied]

Thus, creation of a test program to verify a software application is actually excluded from the TETware system according to the TETware documents, as it requires someone else to supply the test program's code.

The TETware case controller can operate in "build mode", which Applicant believes the Examiner has interpreted as the generation of test cases. If so, Applicant urges that this is an incorrect interpretation. TETware provides a build tool, which is required when build mode is used. However, the build tool merely translates source test cases into executables (Prog, Sec. 3.2). These source test cases are obtained from a list in a TETware scenario.

Support for the amendment of claim 1 is found in the Specification (para [0116]), which explains that the claimed behavioral model is used to model the software application being verified, and is created responsively to the specification of the software application. The behavioral model is then used for purposes of the creation of test programs to test the software application being verified. The scenario file described in the TETware documents at most describes the behavior of the TETware execution engine, TCC, running in execution mode (Prog, Sec. 4.1, and Sec. 4 generally), but says nothing about the software application being verified. The TETware system is thus indifferent to the method by which the test programs it processes were created. The TETware documents do not disclose any components that describe the creation of test programs as claimed in amended claim 1, and are completely silent on modeling the underlying software being tested.

The Examiner has asserted that a behavioral model (scenario file) disclosed in the TETware documents is associated with a focus. The TETware documents are not explicit on this point, although the TETware scenario does have associated data files. However, as the claimed behavioral model is not disclosed in the TETware documents, Applicant cannot agree with the Examiner that the claimed combination of a behavioral model associated with a focus is anticipated by the TETware documents.

The Examiner has further asserted that directives are disclosed in the TETware documents. These directives are instructions to the TETware execution engine regarding execution of existing test cases listed in the TETware scenario, e.g. how to execute the test cases, where the unit under test is located, how often to repeat the test cases, how long to wait for timeouts, etc. In contrast, in amended claim 1, Applicant has claimed directives which control the creation of the test programs themselves. Support for this is found in paragraph [0117]). Here the Specification discloses directives to generate different test suites, which are to be executed by a software application being verified for different purposes, e.g. regression suites, acceptance suites, and full functional test suites. Thus Applicant urges that amended claim 1 now distinguishes test execution directives as described in the TETware documents from test generation directives as claimed herein.

Claims 2, 3, 4, 11, 12, 13, 14.

For the reasons given above in the discussion of claim 1, the claimed model independent directive and model dependent directives are directives to a test generation step. This is not disclosed in the TETware documents, which only provides directives to a test execution operation.

Claims 5, 6, 7, 8, 9.

The variables discussed by the Examiner in the rejection of Claim 5 have nothing to do with the claimed coverage variables. TETware variables are test execution specific variables, e.g. TET_SAVE_FILES, TET_PASS_TC_NAME which refer to execution parameters for running test cases. The claimed coverage variables are specific to the software program being verified, and have meaning only with reference to the program being verified being tested and its behavioral model (Specification, paragraph [0141]).

Claims 15-28.

Independent claim 15 has been amended similarly to claim 1. Claims 15 – 28 are now believed to be allowable for the reasons given in the discussion of claims 1 – 14.

Claim 29.

Claim 29 is believed to be allowable for the reasons given in the discussion of claim 1.

Claim 30.

The subject matter of claim 30 corresponds generally to the disclosure of the TETware documents. However, claim 30 is believed to be allowable as depending from an allowable claim.

Claims 31, 32, 33.

Claims 31 – 33 are believed to be allowable for the reasons given in the discussion of claims 1- 15. The TETware documents do not disclose the generation of an abstract test suite.

Claims 34 – 46.

Claims 34 - 46 are believed to be allowable for the reasons given in the discussion of claims 1 – 14.

Claims 47 – 64

Independent claim 47 has been amended similarly to claim 1. Claims 47 - 64 are believed to be allowable for the reasons given in the discussion of claims 1 – 14.

Claim 65 - 81.

Independent claim 65 has been amended similarly to claim 1. Claim 65 is believed to be allowable for the reasons given in the discussion of claim 1. Claims 66 – 81 are believed to be allowable as depending from an allowable claim, and further, for the reasons given in the discussion of claims 1 – 14.

Rejections Under 35 U.S.C. § 103

Claims 10, 24, 42, 60, and 78, which were rejected under 35 U.S.C. § 103 as being unpatentable over TETware in view of the use of HTML, are believed to be allowable as depending from allowable claims, and in view of the discussion of claims 1 – 14 given above. The TETware documents do not suggest any modifications that would enable it to function in accordance with claims 10, 24, 42, 60, and 78.

CONCLUSION

It is believed that the amendments and remarks presented hereinabove are fully responsive to all the grounds of rejection and objections raised by the Examiner, and that the Application is now in order for allowance. Applicant thanks the Examiner for his thorough consideration of the Application and appreciates the careful analysis of the art cited therein. The Examiner is invited to contact the undersigned at the number below if any questions should arise.

Respectfully submitted,

Charles J. Gross
Reg. No. 52,972

Andrew M. Calderon
Reg. No.38,093

McGuireWoods, LLP.
1750 Tysons Boulevard, Suite 1800
McLean, VA 22102-4215
703-712-5341

\\COM\436350.1